0.导入导出

从txt导入

Vector\* import\_vector\_from\_txt(const char\* filename);

Matrix\* import\_matrix\_from\_txt(const char\* filename);

导出到txt

void export\_vector\_to\_txt(const Vector\* vector, const char\* filename);

void export\_matrix\_to\_txt(const Matrix\* matrix, const char\* filename);

1. 单个向量

显示向量第i个元素

​​void display\_vector\_element(const Vector\* vector, int index);

做向量的切片

​​Vector\* slice\_vector(const Vector\* vector, int\* indices, int num\_indices);

初始化列向量

​​Vector init\_column\_vector(int size, double value);

 绝对值    
Vector abs\_vector(Vector vec);   
Matrix abs\_matrix(Matrix mat);

向量元素的最大值

​​double max\_element\_vector(const Vector vector);

向量元素的最小值

​​double min\_element\_vector(const Vector vector);

向量C1个最大元素位置

​​void top\_c1\_max\_indices(const Vector vector, int c1, int\* indices);

向量C1个最小元素位置

​​void top\_c1\_min\_indices(const Vector vector, int c1, int\* indices);

向量元素排序（从大到小）

​​void sort\_vector\_desc(Vector\* vector);

向量元素排序（从小到大）

​​void sort\_vector\_asc(Vector\* vector);

向量元素取倒数

​​void reciprocal\_elements\_vector(Vector\* vector);

向量所有元素平方

​​void square\_elements\_vector(Vector\* vector);

替换向量的指定元素

​​void replace\_vector\_element(Vector\* vector, int index, double value);

向量归一化（按范数）

​​void normalize\_vector(Vector\* vector, double norm\_type); // norm\_type: 1, 2, or INF

向量除以数

​​void divide\_vector\_by\_scalar(Vector\* vector, double scalar);

向量乘以数

​​void multiply\_vector\_by\_scalar(Vector\* vector, double scalar);

向量排序并返回新向量

​​Vector sort\_vector(Vector vec);

2. 单个矩阵

显示矩阵第i、j个元素

​​void display\_matrix\_element(const Matrix\* matrix, int row, int col);

矩阵的切片：行

​​double\* slice\_matrix\_row(const Matrix\* matrix, int row);

矩阵的切片：列

​​double\* slice\_matrix\_col(const Matrix\* matrix, int col);

读取矩阵的行列相交元素

​​double get\_matrix\_element(const Matrix\* matrix, int row, int col);

矩阵所有元素和

​​double sum\_of\_matrix(const Matrix\* matrix);

矩阵每一行最大值

​​void max\_elements\_per\_row(const Matrix\* matrix, double\* max\_values);

矩阵每一行最小值

​​void min\_elements\_per\_row(const Matrix\* matrix, double\* min\_values);

矩阵元素的最大值

​​double max\_element\_matrix(const Matrix\* matrix);

矩阵元素的最小值

​​double min\_element\_matrix(const Matrix\* matrix);

矩阵元素取倒数

​​void reciprocal\_elements\_matrix(Matrix\* matrix);

矩阵所有元素平方

​​void square\_elements\_matrix(Matrix\* matrix);

替换矩阵的指定列

​​void replace\_column(Matrix\* matrix, const double\* new\_column, int col\_index);

矩阵列向量间的夹角Cos值

​​double\* calculate\_all\_cosine\_similarities(const Matrix\* matrix);

矩阵的转置

​​Matrix\* transpose\_matrix(const Matrix\* matrix);

矩阵乘数

​​Matrix multiply\_matrix\_by\_scalar(const Matrix\* matrix, double scalar);

整行元素都为0的行编号

​​int\* find\_zero\_rows(const Matrix\* matrix, int\* num\_zero\_rows);

生成随机对称矩阵

​​Matrix generate\_random\_symmetric\_matrix(int rows, int cols);

矩阵的范数

double norm\_1\_matrix(const Matrix\* matrix);

double norm\_2\_matrix(const Matrix\* matrix);

double norm\_inf\_matrix(const Matrix\* matrix);

求解上三角矩阵的逆

​​Matrix invert\_upper\_triangular\_matrix(const Matrix\* m);

2.2特征值

幂法（加分项）

反幂法（加分项）

Rayleigh商迭代（加分项）

3. 向量与向量

向量加法

​​Vector add\_vectors(const Vector v1, const Vector v2);

向量减法

Vector subtract\_vectors(const Vector v1, const Vector v2);

向量点除（对应元素相除）

​​Vector elementwise\_divide(const Vector v1, const Vector v2);

向量点乘（对应元素相乘）

​​Vector elementwise\_multiply(const Vector v1, const Vector v2);

夹角余弦值

​​double cosine\_similarity(const Vector v1, const Vector v2);

向量内积

​​double dot\_product(const Vector\* v1, const Vector\* v2);

向量相乘返回矩阵（外积）

Matrix vector\_outer\_product(const Vector\* vec1, const Vector\* vec2);

4. 向量与矩阵

矩阵乘向量

Vector multiply\_matrix\_by\_vector(const Matrix\* matrix, const Vector\* vector);

向量乘以矩阵

Vector\* multiply\_vector\_by\_matrix(const Vector\* vector, const Matrix\* matrix);

5. 矩阵与矩阵

矩阵加法

Matrix add\_matrices(const Matrix\* m1, const Matrix\* m2);

矩阵减法

Matrix subtract\_matrices(const Matrix\* m1, const Matrix\* m2);

ATBA型矩阵相乘

Matrix multiply\_ATBA(const Matrix\* A, const Matrix\* B);

矩阵的点乘（对应元素相乘）

​​Matrix elementwise\_multiply\_matrices(const Matrix\* m1, const Matrix\* m2);

矩阵点除（对应元素相除）

​​Matrix elementwise\_divide\_matrices(const Matrix\* m1, const Matrix\* m2);

AB型矩阵相乘

​​Matrix multiply\_matrices(const Matrix\* A, const Matrix\* B);